Université Ibn Khaldoun – Tiaret
Département d'Informatique
Responsable du module : Khaled ABID

Module : Initiation à l'algorithmique
Niveau : 1ière année LMD
Année : 2020-2021

Fiche de TD n° 5 (les tableaux, les matrices)

Exercice 1 :

Ecrire un algorithme qui déclare et remplisse un tableau contenant les six voyelles de l'alphabet latin.

```
int main() {
      char tab[6] = { 'a', 'e',
'i', 'o', 'u', 'y' };
      int i;
      for (i = 0; i < 6; i++)
            printf("%c ",
tab[i]);
      return 0;
}
```

```
int main() {
      char tab[6];
      tab[0] = 'a';
      tab[1] = 'e';
      tab[2] = 'i';
      tab[3] = 'o';
      tab[4] = 'u';
      tab[5] = 'y';
      int i;
      for (i = 0; i < 6; i++)
            printf("%c ",
tab[i]);
      return 0;
}
```

Exercice 2 :

Ecrire un algorithme permettant à l'utilisateur de saisir un nombre quelconque de valeurs, qui devront être stockées dans un tableau. Une fois la saisie terminée, le programme affichera le nombre de valeurs négative et le nombre de valeur positives.

```
int main() {
      int tab[100];
      int n, i, nbP = 0, nbN = 0;
      scanf("%d", &n);
      for (i = 0; i < n; i++)
            scanf("%d", &tab[i]);

      for (i = 0; i < n; i++) {
            if (tab[i] < 0)
                  nbN++;
            else
                  nbP++;
      }
      printf("Nombre des valeurs positives = %d\n", nbP);
      printf("Nombre des valeurs negatives = %d\n", nbN);
      return 0;
}
```

Exercice 3 :

- Ecrivez un algorithme constituant un tableau, à partir de deux tableaux de même longueur préalablement saisis. Le nouveau tableau sera la somme des éléments des deux tableaux de départ. Par Exemple :

Tableau1 :

5	6	1	8	9	2

Tableau2 :

3	7	1	3	5	4

Tableau à constituer :

8	13	2	11	14	6

- Toujours à partir de deux tableaux précédemment saisis, écrivez un algorithme qui calcule les schtroumpfs des deux tableaux. Pour calculer les schtroumpfs, il faut multiplier chaque élément du tableau1 par chaque élément du tableau2, et additionner le tout. Par exemples :

Tableau1 :

2	5	8	4

Tableau2 :

6	7

Le Schtroumpf sera : $6 * 2 + 6 * 5 + 6 * 8 + 6 * 4 + 7 * 2 + 7 * 5 + 7 * 8 + 7 * 4 = 247$

```
int main() {
      int t1[10], t2[10], t3[10];
      int i, j, sch, n = 10;
      // Lecture des tableau t1 et t2
      for (i = 0; i < n; i++)
            scanf("%d", &t1[i]);
      for (i = 0; i < n; i++)
            scanf("%d", &t2[i]);
      //calcule du tableau t3
      for (i = 0; i < n; i++)
            t3[i] = t1[i] + t2[i];
      //Affichage des tableaux t1, t2 et t3
      for (i = 0; i < n; i++)
            printf("%d ", t1[i]);
      printf("\n");
      for (i = 0; i < n; i++)
            printf("%d ", t2[i]);
      printf("\n");
      for (i = 0; i < n; i++)
            printf("%d ", t3[i]);
      printf("\n");
      // Calcule du schtroumpf
      sch = 0;
      for (i = 0; i < n; i++)
            for (j = 0; j < n; j++)
                  sch = sch + (t1[i] * t2[j]);
      printf("schtroumpf = %d", sch);

      return 0;
}
```

Exercice 4 :

- Ecrivez un algorithme qui renvoie la plus grande valeur dans un tableau en précisant quelle position elle occupe dans un tableau.

```
int main() {
    int tab[10];
    int i, max, pos;
    // Lecture du tableau
    for (i = 0; i < 10; i++)
          scanf("%d", &tab[i]);
    max = tab[0];
    pos = 0;
```

```
    for (i = 1; i < 10; i++) {
        if (max < tab[i]) {
            max = tab[i];
            pos = i;
        }
    }
    printf("max  est %d dans la position %d",max, pos);
    return 0;

}
```

- Toujours que le même principe, écrivez un algorithme permettant à l'utilisateur de saisir les notes d'une classe et renvoie le nombre des notes supérieures à la moyenne de la classe.

```
int main() {
    float notes[30], moyenne, s;
    int i, nb = 0;
    // Lecture des notes
    for (i = 0; i < 30; i++)
        scanf("%f", &notes[i]);
    // Calcule de la moyenne
    s = 0;
    for (i = 0; i < 30; i++)
        s = s + notes[i];
    moyenne = s / 30;
    for (i = 0; i < 30; i++) {
        if (notes[i] > moyenne)
            nb++;
    }
    printf("La moyenne  est %f \n", moyenne);
    printf("Le nombre des notes supérieures à la moyenne est %d",
nb);
    return 0;

}
```

Exercice 5 :

Soit un tableau T à deux dimensions (12,8) préalablement rempli de valeurs numériques. Écrire un algorithme qui recherche la plus grande valeur au sein de ce tableau.

```
int main() {
    int T[12][8];
    int i, j, max;
    // Lecture du tableau
    for (i = 0; i < 12; i++)
        for (j = 0; j < 8; j++)
            scanf("%d", &T[i][j]);
    max = T[0][0];
    for (i = 0; i < 12; i++)
        for (j = 0; j < 8; j++) {
            if (max < T[i][j]) {
                max = T[i][j];
            }
        }
    printf("La plus grande valeur dans T est %d", max);
    return 0;

  }
```

Exercice 6 :

Ecrivez un programme qui affiche le triangle de Pascal en utilisant un tableau de deux dimension de taille NxN.

$$Formule\ de\ pascal: \begin{cases} T[i,0] = 1 \\ T[i,j] = 1 \quad Si\ (i == j) \\ T[i,j] = T[i-1,j] + T[i-1,j-1]\ \ pour\ (1 \leq j < i\ ) \end{cases}$$

```
int main() {
      int T[10][10];
      int i, j;
      for (i = 0; i < 10; i++)
            T[i][0] = 1;
      for (i = 0; i < 10; i++)
            T[i][i] = 1;
      for (i = 2; i < 10; i++)
            for (j = 1; j < i; j++) {
                  T[i][j] = T[i - 1][j] + T[i - 1][j - 1];
            }
      for (i = 0; i < 10; i++) {
            for (j = 0; j <= i; j++)
                  printf("%d ", T[i][j]);
            printf("\n");
      }
      return 0;

   }
```